AMENDMENTS TO THE SPECIFICATION

Please amend the specification as indicated hereafter. It is believed that the following amendments and additions add no new matter to the present application.

In the Title:

Please replace the title with the following new title: APPARATUS AND METHOD FOR GENERATING A SET OF TEST VECTORS USING NONRANDOM FILLING.

In the Abstract:

Please replace the pending abstract with the newly-submitted abstract attached herewith on a separate sheet.

In the Specification:

Please amend the paragraph starting on p. 1, line 8 as follows:

The present invention generally relates to testing devices, and more particularly to an improved test pattern generator for generating test ~~sequences~~ vectors in an automatic test pattern generator.

Please amend the paragraph starting on p. 1, line 13 as follows:

A variety of automatic test equipment (ATE) ~~have~~has long been known for testing electronic circuits, devices, and other semiconductor and electronic products. Generally, automatic test equipment ~~are~~is divided into two broad categories, analog testers and digital testers. As the names imply, analog testers are generally designed for testing analog circuit devices, while digital testers are designed for testing digital circuit devices. Digital testers, as is known, generally include a testing device having a number of internal circuit cards or channels that generate programmably controlled test signals for testing and evaluating a Device Under Test (DUT). More specifically, ATE are programmably controlled to be adapted or configured to testing a variety of devices in a variety of ways. This is achieved by programming ATE inputs to inject a certain signal (or signal transition) and by programming ATE outputs to compare a value to a certain pin or signal line on a DUT. In this regard, a digital tester generally includes a test head whereby electrical signals are input to and output from the tester.

The test head comprises a number of connectors, each defining a channel, which may be connected via cable or otherwise to a device under test. The electronics within the digital tester may then input and output signals to/from a DUT via the test head.

Please amend the paragraph starting on p. 3, line 6 as follows:

A test vector or test pattern, as generated by a test generator, is a string of n logic values (0, 1, or don't care – X) that are applied to the n corresponding primary inputs (PIs) <u>and/or scan input (SI) logic values </u>of a circuit at the same time frame<u>, usually accompanied by m primary outputs (POs) and/or scan output (SO) logic values that represent the expected circuit response</u>. A test sequence is a series of test vectors applied to a ~~sequential~~ circuit <u>(typically a sequential circuit) </u>in a specific order to detect a target fault. The first vector in the test sequence assumes the circuit to be in a completely unknown state. A test set is an unordered set of test <u>vectors (for combinational circuits) or test </u>sequences<u> (for sequential circuits)</u>. For combinational circuits with combinational faults, there is no need for test sequences and the test set consists of an unordered set of test vectors. <u>Stated another way, combinational circuits are tested with test sets containing degenerate test sequences that contain only one test pattern each. </u>Test length refers to the number of test vectors in a test set.

Please amend the paragraph starting on p. 6, line 20 as follows:

To achieve the advantages and novel features, the present invention is generally directed to an improved automatic test pattern generator for generating test patterns that are used by an integrated circuit testing device. In accordance with one aspect of the invention, a method is provided for generating a set of test ~~sequence~~<u>vector</u>s for testing an integrated circuit, each test ~~sequence~~<u>vector</u> of the set of test ~~sequence~~<u>vector</u>s containing a plurality of bits defining test inputs for the integrated circuit. The method includes the steps of defining a list of faults for the integrated circuit, and generating at least one test ~~sequence~~<u>vector</u> that defines values for those inputs necessary to detect at least one target fault selected from the list of faults, the values comprising only a portion of the bits of the at least one test ~~sequence~~<u>vector</u>, wherein a remainder of the bits in the at least one test ~~sequence~~<u>vector</u> are unspecified bit positions. The method further includes the step of setting the

values of a plurality of the unspecified bit positions using a non-random filling methodology.

Please amend the paragraph starting on p. 7, line 11 as follows:

In accordance with the invention, the step of setting the values of a plurality of the unspecified bit positions may be carried out in various ways. For example, the step of setting the values of the plurality of the unspecified bit positions may be performed by setting each of the plurality of the unspecified bit positions to a one. Likewise, the step of setting the values of the plurality of the unspecified bit positions may be performed by setting each of the plurality of the unspecified bit positions to a zero. Further, the step of setting the values of the plurality of the unspecified bit positions may be performed by setting each of the plurality of the unspecified bit positions to a repeating or periodic pattern of ones and zeros. Further still, the step of setting the unspecified bit positions may be performed by setting a plurality of the unspecified bit positions in accordance with a random filling methodology, while another plurality of the unspecified bit positions are set in accordance with a non-random filling methodology, such as one of the methodologies described above. In addition, all unspecified bit positions may be set to the last specified value in the test ~~sequence~~vector (*i.e.*, extending the last specified value throughout the unspecified bit positions).

Please amend the paragraph starting on p. 10, line 8 as follows:

The test generator 102 uses the data provided by ~~from~~ the device model 108 and generates a compacted set of test vectors. Once the compacted set of test vectors is created, it is transferred to the ATE 104, where the compacted set may be used over and over to test DUTs 116.

Please amend the paragraph starting on p. 14, line 1 as follows:

Reference is now made to FIG. 4, which illustrates a more practical implementation of the scan chains and scan register of the preferred embodiment. In this regard, instead of providing a separate register to comprise the scan register 312, typically sequential logic already embedded within the circuit chip is utilized. For example, and again for purposes of illustration, assume flip flops (*e.g.*, 420) are

provided in an integrated circuit chip, and are functionally configured to operate in a certain manner. Test vector values may be shifted into these registers via, for example, multiplexers 422. In this regard, a multiplexer 422 may have 2 inputs: one for receiving an input from the functional logic 424 provided on the chip, and one for receiving input values from a scan input 427 provided in connection with the testing configuration of the chip. A scan enable line 426 may be provided as a multiplexer select, to select which of the two inputs is routed through the multiplexer 422 to the flip flop 420. Once the various bit values of the test vector 306 have been shifted into the scan chain 412, the scan enable line 426 may be set to propagate the appropriate bit values of the scan chain 412 to the various sequential circuit devices (*e.g.*, 420). As will be understood, a clock line (denoted as scan clock) 428 may be toggled to clock in the various bit values through the respective sequential circuit components comprising the scan chain (or scan register) 412. In this way, the various outputs of the sequential circuit components may be controllably set in order to test the combinational logic 408 of the integrated circuit chip. In this regard, it is assumed that the functional logic of an integrated circuit chip will comprise a combination of sequential and combinational logic, which may be organized in various layers (*e.g*, a layer of sequential logic, then a layer of combinational logic, then a layer of sequential logic, another layer of combinational logic, *etc*.). Any given "layer" of combinational logic may be tested by controlling the values directed to the inputs of that combinational logic, in a manner described above and illustrated in connection with FIG. 4, and observing its outputs. The outputs of the combinational logic components may then be directed to one or more output scan chains, which then may be shifted serially out of the integrated circuit chip for evaluation by the testing equipment. In this regard, and as illustrated in FIG. 3, a separate output scan register may be formed within the chip, or alternatively, the output scan chain utilize the same sequential registers as the input scan chain.

Please amend the paragraph starting on p. 16, line 21 as follows:

Now referring to FIG. 5B, if the output of the AND gate ~~538~~537 were to be tested for a "stuck at" "1" fault, at least one of the input lines must be set to "0". The two remaining input lines could be set to don't care values (e.g., "X"). In this regard, the three input bits 540, 541, and 542 may comprise the values "0XX" "X0X", or

"XX0", as any of these three values would properly test the stuck at "1" fault condition on line 544. An output bit position 546 is associated with the signal line 544 in order to read and test the value output from AND gate ~~538~~537. Thus, with the input of "0XX" as illustrated in FIG. 5B, the output of bit position 546 should be a 0, if the AND gate ~~538~~537 is working properly. If, however, there is a stuck at "1" fault at the output of AND gate ~~538~~537, the bit position of 546 would be a 1.

Please amend the paragraph starting on p. 17, line 9 as follows:

Reference is now made to FIG. 5C. In similar fashion, a stuck at "0" fault at the output of AND gate ~~538~~537 may be tested by setting bit positions 540, 541, and 542 all to "1". The output bit position 546 may then be monitored to determine~~d~~ whether it is a "1" or "0" (in response to the "111" inputs). A "1" in the bit position 546 indicates proper operation of the AND gate ~~538~~537. However, a "0" indicates a stuck at "0" fault on line 544.

Please amend the paragraph starting on p. 24, line 4 as follows:

As between random versus non-random filling, it is believed that random filling may realize a slightly greater fault coverage. However, non-random filling can realize other benefits. For example, test sets generated by using non-random fill methodologies may be much more effectively compressed, thereby requiring less memory for storage. It will be appreciated that non-random filling may be implemented in any of several ways. For example, all unspecified bit positions may be set to a value of one. Likewise, all unspecified bit positions may be set to a value of zero. In addition, all unspecified bit positions may be set to the last specified value in the test ~~sequence~~vector (*i.e.*, extending the last specified value throughout the unspecified bit positions). For example, the vector "0XX1X0XXX0X1XXX" would become "000110000001111" using such an "extended" filling methodology.